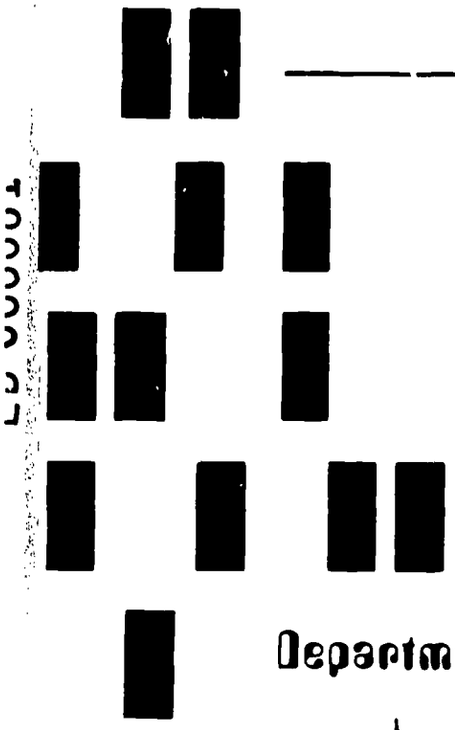
DOCUMENT RESUME

ED 066 881 EM 010 168

AUTHOR	Billingsley, Ray; Wilson, Stanley
TITLE	Program and Model Documentation Standards.
INSTITUTION	Texas A and M Univ., College Station. Texas Agricultural Experiment Station.
REPORT NO	R-71-1
PUB DATE	Feb 71
NOTE	23p.; Agricultural Economics series
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Programs; *Documentation; Guides; *Models; *Specifications

ABSTRACT

The purpose of a computer program or model documentation is to provide the details that will aid others to use the program, and, more importantly, aid those who desire to modify or revise the program. This pamphlet presents specifications for such documentation, first offering a complete outline of all the information that should appear. Each section of documentation--program identification and background, user documentation, and programer documentation--is then described in detail. Examples are appended. (Author/SH)



Departmental Program and Model Documentation

PROGRAM AND MODEL DOCUMENTATION STANDARDS

71-1

Texas A&M University
The Texas Agricultural Experiment Station
H. O. Kunkel, Acting Director

AGRICULTURAL ECONOMICS PROGRAM

AND MODEL DOCUMENTATION

71-1

PROGRAM AND MODEL DOCUMENTATION

STANDARDS

by
Ray Billingsley
and
Stanley Wilson

February 1971

Table of Contents

INTRODUCTION

The purpose of a computer program or model documentation is to provide the details which will aid others to use the program but, more importantly, aid those who desire to modify or revise a program. Often a potential user cannot effectively utilize all the features of a program if he does not have an understanding of how the program operates. For more sophisticated programs, particularly those which simulate models of economic organizations and systems, any use at all is impossible without a substantial and detailed understanding of the program's algorithm, mathematical processes and assumptions. In any case, a simple explanation of where on the data card to enter the various values is seldom sufficient.

For these reasons, two types of documentation are useful. The first is for the person who wants to use the program as it exists. The second is for the person who would like to modify the program in order to meet his specific requirements or simply to improve the program. For these reasons, the program documentation specifications have been divided into USER DOCUMENTATION and PROGRAMMER DOCUMENTATION. Carefully written programs should also be generously endowed with comment cards which aid in program modification.

A number of programs or linear programming models have been written over the last several years which have not been documented. Substantial time, both professional and computer, has been expended to debug these programs and models, but after they were used for a particular project it was not possible for another person to use them again without virturally starting over. This results in a wasteful use of professional and computer time and incapacitates further use of the programs and models developed. Even if the existence of a similar program or model is known, its use may be impaired when the documentation does not exist or is so sketchy that modifying the existing program requires a great deal of effort. Often a program or model has wide possibilities for application by others and in some cases this may be more useful than its application to a particular problem.

The development of program and model documentation standards is intended to generate a class of uniform reports specifically devoted to program and model use and to provide a professionally recognized medium for displaying the work of those who expend efforts in building programs and computer simulations. The standards of documentation worked out in this document are designed so that the program or model can be understood both by the casual as well as the sophisticated user. Examples of program and model documentation are given in Appendices A and B respectively.

To achieve uniformity, an ordered listing of the material included in each documentation is required. This listing, called "CONTENTS OF PROGRAM AND MODEL DOCUMENTATION," is on the next page and should form the introduction and table of contents of the documentation. If a particular element in the documentation is not applicable, the author should state "none" after the element name in the body of the documentation and place "XXX" in the page reference space on the contents page.

CONTENTS OF PROGRAM AND MODEL DOCUMENTATION

* If a particular element is not contained in this documentation place XXX in this column.

DOCUMENTATION ELEMENTS

I. PROGRAM IDENTIFICATION AND BACKGROUND

1. PROGRAM NAME
2. DOCUMENTATION NUMBER
3. AUTHOR OF DOCUMENTATION
4. PROGRAMMER
5. ORIGIN OF PROGRAM
6. LANGUAGE/COMPUTER
7. DATE

Each computer program or model should be given a different name. If the name is not indicative of the function of the program, a short statement should be included clarifying this. The documentation number identifies the year and the chronological order of the series for that year. The author of the documentation is usually a professional who initiates the program by having some process or model which he would like to see embodied in a program. He communicates this process or model in the form of an algorithm and/or flowchart to the programmer, who prepares the program, codes and debugs it. If the programmer produces part of the algorithm, he should be listed as co-author of the documentation. If an existing program written by another is modified, then the author should make reference to the original author and the original documentation, if known, in the section called "Origin of Program." If the program is original with the author, then the word "Original" should appear after the phrase "Origin of Program." If the author modifies his own program, he should make reference to the original program and documentation. The language refers to the computer language in which the program is written. When the documentation is separated from the program this information is not always obvious. The date refers to the date on which the program became operational.

II. USER DOCUMENTATION

1. GENERAL ABSTRACT: A short (not more than one page) description of the purpose of the program or model and how the program operates is required. This abstract should be directed to

professionals in the authors field and should indicate the functions which the program serves in that field. This abstract, to be suitable for indexing and use in information storage and retrieval systems, should be less then 200 words.

2. OPERATIONAL USE: The following information should be provided to facilitate use of a program or model.

    Data requirements/options
    Input specifications/format
    Jobstream

A program may have several options and these options may determine which variables must be assigned values by inclusion as part of the data and which variables have their values calculated by the program. Data requirements, in turn, will determine the exact input specifications, that is the sequence in which the data cards are read in and whether a particular variable is assigned a numerical value in the input or is left blank. Frequently these specifications will consist of the data card formats alone. Particular attention should be given to situations in which disk files or tapes are to be utilized. The jobstream represents the sequence of control cards, program cards, data cards, etc. necessary to run the program.

III. PROGRAMMER DOCUMENTATION

1. GENERAL DESCRIPTION: The general description is intended to provide adequate information to convey what the program is intended to do and how it does it. References to more thorough treatments of the mathematical model than are included in the programming documentation are encouraged. The specification of the following items will constitute a reasonable general description of the program.

    Purpose
    Assumptions and capabilities
    Explanation of algorithm
    Mathematical description
    References

The purpose consists ofa short statement of what the program is supposed to do. This is followed by any special assumptions on which the model is based. These assumptions indicate any special capabilities or limitations of the program or model. The algorithm consists of an ordered series of steps by which the computer accomplishes the task or simulates the model.

Emphasis should be placed on the order in which the operations are carried out. Performing the task may depend more on the sequence in which the operations are done and less on each particular operation (which may be fairly simple). The contribution of the author may consist more in designing an algorithm than stating a new principle or a mathematical proof.

The mathematics of the program may be so simple or well known that this section is unnecessary. Alternately the program may be based on a mathematical proof or demonstration. In this case a clear, complete demonstration of the proof or reference to more thorough treatment of the mathematical model would be helpful.

The author should refer to any publications which will be useful to the reader or which can be used to go deeper into various aspects of the subject. If the program embodies techniques or principles discovered by another, the credit should be given to him. Both references to publications on techniques (such as books on linear programming) and subject area publications (such as books on economics) may be included. The form of reference should be standard bibliographical form and each entry should be assigned a number. In this way. referring to page 126 of the third entry in the bibliography, he can simply put [3, p. 126] in or after the sentence most closely associated with the reference. The references should certainly include any manuscripts prepared using this program or model.

2. PROGRAM DESCRIPTION: For the main program and each of the major sub-programs the following items are required:

    Abstract
    Variable list
    Subroutines and functions called
    Flowchart

The abstract should identify the purpose of the program. It should be directed to programmers and should indicate the role the main program and each sub-program plays in accomplishing the functions of the program or model. The variable list should include the names and interpretations of major variables used in the program. It is generally useful to distinguish between subscripted and simple variables. Listing the subroutines and functions called from the program generally provides useful reference information. A general flowchart of the logic of a program is useful. Simple, liberally commented descriptions may be preferable in some cases to detailed flowcharts. The flowchart should be coordinated with the listing in that the numbers of cruical statements in

the listing should appear at the corresponding point on the flowchart. A programmer wishing to modify the original program can obtain a detailed flowchart by using a flowchart routine. Several are available.

3. PROGRAM LISTING: A reproduction of the computer listing is usually the best way to present the program listing. It not only eliminates the possibility of typographical error, but also conveys a better picture of the precise nature of the program.

4. SAMPLE PROGRAM RUN:

    Input File/Sample Data
    Output File

    A sample run in which both the input file and the output file are listed may be useful. Additional annotation of these items may also be very useful. Many programs do not print out the input data before any results are calculated and/or printed. An image of this part of the input is often the easiest way to show the sample input data. This is particularly true if values of certain integers in the input determine which program options are used or how the input data is interpreted (for example which element in an array a given numerical value refers to). A sample output should be included when there is the possibility that it would be instructive to the user. A copy of the output pages alone may not be sufficient. Often some explanation of how to interpret various items in the output is vital. If the program does batch processing (more than one run of the program and introducing changes in a model's parameter or parameters), then a significant relationship may exist between a particular value in the input file and a particular value in the output file. For example, a given change in a certain model parameter may have a strong effect on a certain output value. If so, these relationships should be pointed out in describing the output file. Similarly, the formats of output files should be described carefully, particularly where information is sorted on tapes for future use.

APPENDIX A

CONTENTS OF PROGRAM AND MODEL DOCUMENTATION

* If a particular element is not contained in this documentation place XXX in this column

II. USER DOCUMENTATION

1. GENERAL ABSTRACT

SEP PRICE is a computational device designed for use in NUPLEX planning and analysis. It generates a series of price and quantity changes given the price elasticity of demand and computes the SEP PRICE appropriate for each separable segment so that the average price received for the quantity of product sold is equal to the expected market price. SEP PRICE corresponds to the marginal revenue in the conventional imperfect competition model.

2. OPERATIONAL USE

Data Requirements: The program requires the elasticity of demand -- E, a beginning quantity -- Q, beginning average price -- P, a change in quantity -- DELQ, and the number of times Q is to be changed by DELQ -- N.

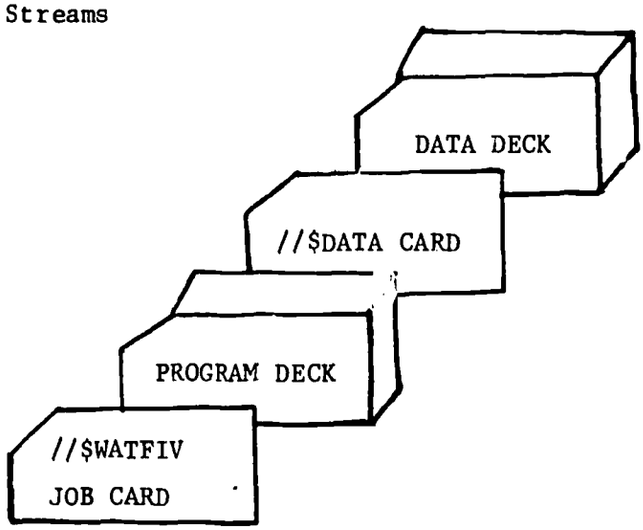
Options: If the user wishes to move out along the demand curve, that is if the quantity is to be increased, then DELQ is entered with a plus sign or no sign. If the user wishes to move back along the demand curve, that is if the quantity is to be decreased, then DELQ is entered with a minus sign. In either case the program makes the sign of the change in Price the opposite of the sign of the change in Quantity.

Input Specifications/Format: The first card or cards in the input file are the description cards. The first four spaces on each description card should be left blank except for the last description card. On the last description card a "1" should be entered in Column 1. The user may have as many description cards as he wishes. After the description card comes the data card.

Data Card Form.	Symbol	Spacing	Format
	E	1-7	F7.4
	DELQ	10-24	F15.6
	Q	25-44	F20.4
	P	45-59	F15.4
	N	78-80	I 3

Limitations: The program is limited to one elasticity value per run and one beginning quantity and price. Thus the user gets only one section of a demand curve on a run.

Job Streams



III. PROGRAMMER DOCUMENTATION

1. GENERAL DESCRIPTION

Purpose: The agricultural output of NUPLEX for any given crop could be such a large proportion of U.S. output that the output of NUPLEX would have a significant impact on the market price of the product and therefore on the average revenue. For that reason, sound economic analysis would indicate that this effect must be taken into consideration. This can be done using the separable programming feature of MPS/360.

The purpose of SEP-PRICE is to prepare input data for MPS/360 for separable programming. MPS/360 separable programming allows functions to be broken up into short ranges, and the slope of the line connecting the end points of each section is used to approximate the slope of the function over that section. The SEP-PRICE program breaksup the demand curve into small sections and calculates the SEP-PRICE appropriate for each separable segment so that the average price received for the quantity of product used is equal to the expected market price given the elasticity of demand, the size of the change in quantity, a beginning quantity and price.

Assumptions: The model assumes that each small section of the demand curve can be approximated accurately enough to be useful in deriving price changes from given changes in the quantity.

Algorithm: See Flowchart.

Mathematical Description: The formula used to calculate the change in price is:

$$DELP = [DELQ * P]/[Q * E]$$

where,

E = elasticity of demand,
P = average price,
Q = quantity, and
DELQ = change in quantity,
DELP = change in price

SEP-P is the separable price appropriate for each separable segment so that the average price received is equal to the expected market price.

References: (1) Mathematical Programming System/360, H-20-0476-1, IBM.

2. PROGRAM DESCRIPTION

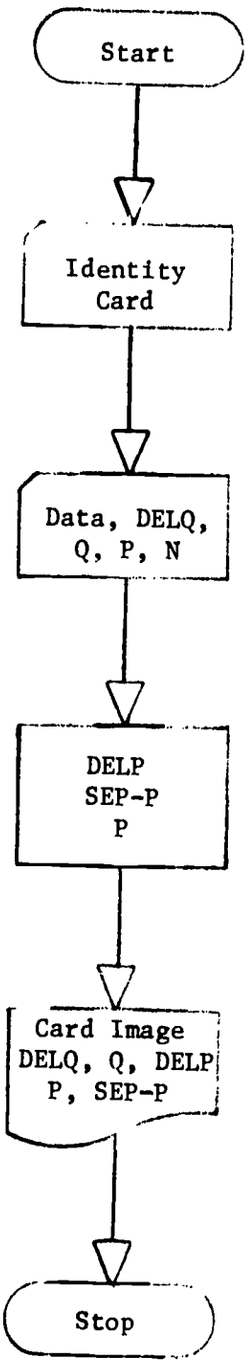
Program Abstract: SEP-PRICE has no subroutines or functions and calls no system subroutines. The GENERAL ABSTRACT provides a sufficient abstract in this case.

Variable List: See mathematical description.

Subroutines/Functions called: None.

Flow Chart: See next page.

Flowchart


Start
Identity Card
Data, DELQ, Q, P, N
DELP SEP-P P
Card Image DELQ, Q, DELP P, SEP-P
Stop

3. PROGRAM LISTING

```
      DIMENSION DES(19), P(100)
      WRITE(6,120)
  120 FORMAT(1H1,20X,28H - PROBLEM IDENTIFICATION - )
    5 CONTINUE
      READ(5,121) IFLAG, (DES(N),N = 1,19)
  121 FORMAT(I1,3X, 19(A4) )
      WRITE (6,122) IFLAG, (DES(N),N = 1,19)
  122 FORMAT(1X,I1,2X,19(A4) )
      IF( IFLAG .EQ. 0 ) GO TO 5
      READ(5,100) E,DELQ,Q,P(1),N
  100 FORMAT(F7.4,2X,F15.6,F20.4,F15.4,18X,I3 )
      WRITE(6,123) E,DELQ,Q,P(1),N
  123 FORMAT(1H0,F7.4,2X,F15.6,F20.4,F15.4,18X,I3)
      WRITE(6,115)
  115 FORMAT(1H1,10X,5HDEL Q,22X,1HQ,21X,5HDEL P,22X,1HP,14X,5HSEP-P)
      DO 10 I = 1,N
      SI = I
      M = N - 1
      SEPP = SI * P(I)
      DO 20  J =  1,M
      IF( (I - J) .LE. 0 ) GO TO 25
      SEPP = SEPP - P(I - J)
   25 CONTINUE
   20 CONTINUE
      DELP =((DELQ * P(I)) / (Q * E)) * (-1.0)
      WRITE(6,110) Q,P(I), DELQ, DELP, SEPP
      Q = Q + DELQ
      P(I + 1) = P(I) + DELP
   10 CONTINUE
  110 FORMAT(21X,1H ,2X,F20.6,2X,1H ,24X,1H ,2X,F20.6,2X,1H /4X,
     1F15.6,2X,1H ,24X,1H ,2X,F20.6,2X,1H ,24X,1H ,2X,F10.5)
      STOP
      END
```

4. SAMPLE PROGRAM RUN

Input File/Sample Data: E = 0.14, DELQ = 90, Q = 9000, P(1) = 4.65, N = 15 for this sample run. Data is not reproduced as part of the output. The description cards are reproduced but not shown here because their contents are self explanatory.

Output File:

DEL Q	Q	DEL P	P	SEP-P
	9000.000000		4.650000	
90.000000		-0.332143		4.65000
	9090.000000		4.317857	
90.000000		-0.305365		3.98571
	9180.000000		4.012492	
90.000000		-0.280987		3.06962
	9270.000000		3.731504	
90.000000		-0.258773		1.94567
	9360.000000		3.472732	
90.000000		-0.238512		0.65179
	9450.000000		3.234220	
90.000000		-0.220015		-0.77928
	9540.000000		3.014204	
90.000000		-0.203113		-2.31938
	9630.000000		2.811090	
90.000000		-0.187656		-3.94429
	9720.000000		2.623434	
90.000000		-0.173508		-5.63321
	9810.000000		2.449926	
90.000000		-0.160546		-7.36830
	9900.000000		2.289380	
90.000000		-0.148661		-9.13432
	9990.000000		2.140718	
90.000000		-0.137755		-10.91825
	10080.000000		2.002963	
90.000000		-0.127740		-12.70908
	10170.000000		1.875222	
90.000000		-0.118535		-14.49746
	10260.000000		1.756687	
90.000000		-0.110068		-16.27548

The above is an example of how the output looks, The first 9000 units can be sold at $4.65. If an additional 90 units are sold, the average price declines to $4.34. In order for the average price to be $4.34, the user of MPS/360 must have the first 9000 units sell for $4.65 and the additional 90 units sell for $4.03, the separable price. If the sales go up to 9180 units, then average price declines to $4.05. The first 9000 units must sell for $4.65, the next 90 sell for $4.03 and the next 90 sell for $3.17 in the input data for MPS/360.

APPENDIX B

CONTENTS OF PROGRAM AND MODEL DOCUMENTATION

* If a particular element is not contained in this documentation place XXX in this column.

II. USER DOCUMENTATION

1. ABSTRACT: This model is designed for class room use to illustrate the basic elements of linear programming

2. OPERATIONAL USE: Detailed information available in the MPS/360 Linear Program User's Manuals, [2].

III. PROGRAMMER DOCUMENTATION

1. GENERAL DESCRIPTION: The linear programming problem used here for illustration is based on a hypothetical farm situation. The objective is to maximize farm income.

Purpose - To illustrate use of LP

Assumption and Capabilities - Maximize farm income

Algorithm - None

References

[1] Billy G. Freeman and Curtis F. Lard, "A Users Guide to Linear Programming and the IBM MPS/360 Computer Routine," Departmental Technical Report Number 70-2, Department of Agricultural Economics and Sociology, Texas A&M University, June 1970.

[2] "IBM Linear Program Users Manuals" (H20-0291) and (H20-0476-0).

2. MODEL DESCRIPTION

Abstract - See section II.1

Variable list -

Row Name	Explanation
1OBJ	Objective Function 1
2OBJ	Objective Function 2
CHRow 1	Change Row
1CROP	Cropland
2PAST	Pasture
3CTAL	Cotton allotment
4FGB	Feed grain base
5LBR	Labor 1
6LBR	Labor 2
7LBR	Labor 3
8LBR	Labor 4
9OTZ	Oats grazing transfer
10OST	Steer selling

Column Name	Explanation
1OTG	Oats grain
2OGF	Oats graze
3GSG	Grain sorghum
4COT	Cotton
5COW	Cow calf
6STS	Steers
7STS	Steers sell
8LAB	Labor Buying 1
9LAB	Labor Buying 2
10LAB	Labor Buying 3
11LAB	Labor Buying 4

Right Hand Side

PO1
PO2

Subroutine and Functions called - none

Flow chart - none

3. PROGRAM LISTING: See Page 19

4. SAMPLE PROGRAM RUN:

Input File/Sample Data: See page 20.

Output File: None

Control Program

```
PROGRAM
INITIALZ
MOVE(XDATA, 'MAX')
MOVE(XPBNAME,'PBFILE')
MOVE(XOBJ,'2OBJ')
MOVE(XRHS,'PO1')
CONVERT('SUMMARY')
BCDOUT
SETUP('MAX')
PICTURE
TRANCOL
PRIMAL
SOLUTION
EXIT
PEND
```

These statements are necessary for each problem. MAX, 2OBJ, and PO1 are arbitarary names which are peculiar to this problem only.

CONVERT checks the input data and converts it to PROBFILE.

BCDOUT causes data to be printed.

SETUP is necessary, MAX is optimal (refer to discussion of objective function).

Causes optimal solution to be computed.

Causes solution to be printed.

EXIT and PEND signals the end of program.

RCDOUT, Data Deck

```
NAME          MAX
ROWS
 N  OBJF
 N  CHROW1
 L  1CROP
 L  2PAST
 L  3CTAL
 L  4FGB
 L  5LBR
 L  6LBR
 L  7LBR
 L  8LBR
 L  9OTZ
 L  10ST
 N  20BJ
COLUMNS
    10TG      OBJF          14.00000   1CROP          1.00000
    10TG      7LBR            .50000   8LBR            .50000
    10TG      20BJ          14.00000
    20GZ      OBJF        - 10.00000   1CROP          1.00000
    20GZ      7LBR            .50000   8LBR            .50000
    20GZ      9OTZ        -   4.00000   20BJ        - 10.00000
    3GSG      OBJF          20.00000   1CROP          1.00000
    3GSG      4FGB           1.00000   5LBR           1.50000
    3GSG      6LBR            .50000   7LBR            .60000
    3GSG      8LBR            .60000   20BJ          20.00000
    4COT      OBJF          16.00000   1CROP          1.00000
    4COT      3CTAL          1.00000   5LBR           1.00000
    4COT      6LBR           2.50000   7LBR            .50000
    4COT      8LBR            .50000   20BJ          16.00000
    5COW      OBJF          18.00000   2PAST          5.00000
    5COW      5LBR           8.00000   6LBR           2.50000
    5COW      7LBR           2.20000   8LBR           9.00000
    5COW      20BJ          18.00000
    6STR      OBJF        - 131.00000  7LBR           2.70000
    6STR      8LBR           1.90000   9OTZ           3.50000
    6STR      10ST        -   6.60000  20BJ        - 131.00000
    7STS      CHROW1         1.00000   10ST           1.00000
    7STS      20BJ          35.00000
    8LAB      OBJF        -   1.50000  5LBR        -   1.00000
    8LAB      20BJ        -   1.50000
    9LAB      OBJF        -   1.50000  6LBR        -   1.00000
    9LAB      20BJ        -   1.50000
    10LA      OBJF        -   1.50000  7LBR        -   1.00000
    10LA      20BJ        -   1.50000
    11LA      OBJF        -   1.50000  8LBR        -   1.00000
    11LA      20BJ        -   1.50000
RHS
    P01       1CROP         150.00000  2PAST          50.00000
    P01       3CTAL          50.00000  4FGB           50.00000
    P01       5LBR         1030.00000  6LBR          780.00000
    P01       7LBR          530.00000  8LBR          780.00000
    P02       1CROP          75.00000  2PAST          25.00000
    P02       3CTAL          25.00000  4FGB           25.00000
    P02       5LBR        -  25.75000  6LBR        -  19.50000
    P02       7LBR        -  13.25000  8LBR        -  19.50000
ENDATA
```